MULTIMEDIA UNIVERSITY

STUDENT ID NO

VENUE : _____

SEAT NO : _____

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2016/2017

TCP 1101 – PROGRAMMING FUNDAMENTALS

(All sections / Groups)

15 OCTOBER 2016
9.00 a.m - 11.00 a.m
(2 Hours)

Question	Marks
1	
2	
3	
4	
Total	

INSTRUCTIONS TO STUDENTS

1. This Question paper consists of 13 pages with 4 Questions only.
2. Attempt All Questions. The distribution of the marks for each question is given.
3. Write all your answers in this Question paper.

Question 1

Unless otherwise stated, every code segment provided in every question below is contained within a `main()` function, complete with the directive `#include <iostream>`, `#include <vector>`, and the statements `using namespace std;` and `return 0;`
Write the output as the result of execution of the following code segments. [10 marks]

(a)

```
double a = 3.5;
int b = 4;
int temp;

cout << a + b << endl;

temp = a;
a = b;
b = temp;

cout << "b = " << b << endl;
```

(b)

```
for( int x = 0; x <= 10; x++ )
{
  cout << x << " ";
  if ( x / 3 > 1 )
  {
    x = x + 1;
  }
}
```

(c)

```
int x, y, p, i;

x = 3; y = 4; i = 1; p = 1;
do
{
  p = p * x;
  i = i + 1;
}
while ( i <= y );
cout << p << endl;
```

(d)

```
int a, b, c;

a = 10; b = 3;

c = a;
while ( c >= b )
{
  c = c - b;
}
cout << "c = " << c << endl;
```

Continued...

(e)

```
#include <iostream>
using namespace std;

double Average(int arr[], int size)
{
  int i, sum = 0;
  double avg;

  for (i = 0; i < size; ++i)
  {
            sum += arr[i];
  }

  avg = sum / size;
  return avg;
}

int main()
{
      int a[3], avg;

      a[0] = 6;
      a[1] = 6;
      a[2] = 1;

      cout << Average(a,2) << endl;

      return 0;
}
```

(f)

```
#include <iostream>
using namespace std;

string dosomething(string &s)
{
    int start = 0;
    int end = s.length();
    string temp = s;

    while (start < end)
    {
        end--;
        temp[start] = temp[end];
        start++;
    }
    return temp;
}

int main()
{
    string s1 = "MMU";

    cout << dosomething(s1) << endl;

    return 0;
}
```

Continued ...

(g)

```
#include <iostream>
using namespace std;

class Account{
    public:
        Account();
        double get_balance();
    private:
        double balance;
};

int main()
{
   Account myacc;
   cout << "My account balance is " <<
myacc.get_balance() << endl;

   return 0;
}

Account::Account() : balance(1000)
{
}

double Account::get_balance()
{
    return 1.1 * balance;
}
```

(h)

```
int a = 5, b = 1, c = 3;
int *p1 = &c;
int *p2 = &a;
p2 = p1;
*p1 = 4;
cout<< b + *p1 + *p2;
```

(i)

```
char name[] = "edward";
char* p = name;
p += 2;
cout<< p <<endl;
```

(j)

```
int a[] = {1, 3, 5, 7, 9, 11, 13, 15};
int *p = a;
(*p)++;
cout<< p[0];
```

Question 2

(a) Write a complete C++ program which converts a binary number to a decimal number. Sample runs are shown below. Assume there will not be any user input errors. You can use the `pow( int x, int y )` function in <cmath> to calculate x^y. [5 marks]

Continued...

Sample Run 1

```
Enter the number of digits of a binary number ( > 0 ): 4
From left to right, enter digit 4: 1
From left to right, enter digit 3: 0
From left to right, enter digit 2: 1
From left to right, enter digit 1: 1
Value in decimal is 11
```

Sample Run 2

```
Enter the number of digits of a binary number ( > 0 ): 3
From left to right, enter digit 3: 1
From left to right, enter digit 2: 1
From left to right, enter digit 1: 0
Value in decimal is 6
```

Continued...

(b) Write a complete C++ program which implements a repeating four-item menu. A sample run is shown below. Include simple input error checking logic to ensure that choice is valid. Assume that user will only enter integers. [5 marks]

Sample Run 1

```
1. Item A
2. Item B
3. Item C
4. Item D
Choice  (Enter 9 to quit) : 1
Item A has been chosen.

1. Item A
2. Item B
3. Item C
4. Item D
Choice  (Enter 9 to quit) : 3
Item C has been chosen.

1. Item A
2. Item B
3. Item C
4. Item D
Choice  (Enter 9 to quit) : 5
Your choice must be between 1 and 4.

1. Item A
2. Item B
3. Item C
4. Item D
Choice  (Enter 9 to quit) : -2
Your choice must be between 1 and 4.

1. Item A
2. Item B
3. Item C
4. Item D
Choice  (Enter 9 to quit) : 9
End program.
```

Continued...

Continued...

Question 3

(a) Write the function definition `swap()` to swap between two values that uses call-by-reference only if the first parameter is **less** than the second parameter. [2 marks]

```
#include <iostream>
using namespace std;

int main()
{
   int x, y;

   cin >> x >> y;

   cout << "Before swap " << endl;
   cout << "x = " << x << ", y = " << y << endl;
   swap(x,y);
   cout << "After swap "<< endl;
   cout << "x = " << x << ", y = " << y << endl;

   return 0;
}
```

Sample output:

1 2 Before swap x = 1, y = 2 After swap x = 2, y = 1	2 1 Before swap x = 2, y = 1 After swap x = 2, y = 1

Continued...

(b) The following code is supposed to show the average quiz mark for each student. Write the function definition for the function `compute_average()`. The function will use the data in multidimensional array `grade[][]` to calculate the average mark and store it in `student_average[]` array [4 marks]

```
#include <iostream>
using namespace std;

const int NO_STUDENT = 4, NO_QUIZ = 3;

void compute_average(int grade[][NO_QUIZ], double
student_average[]);

int main()
{
   int grade[NO_STUDENT][NO_QUIZ];
   double student_average[NO_STUDENT];

   // Student 1 has mark 10, 10, 10
   grade[0][0] = 10;
   grade[0][1] = 10;
   grade[0][2] = 10;

   // Student 2 has mark 2, 0, 1
   grade[1][0] = 2;
   grade[1][1] = 0;
   grade[1][2] = 1;

   // Student 3 has mark 8, 6, 9
   grade[2][0] = 8;
   grade[2][1] = 6;
   grade[2][2] = 9;

   // Student 4 has mark 8, 4, 10
   grade[3][0] = 8;
   grade[3][1] = 4;
   grade[3][2] = 10;

   compute_average(grade, student_average);

   for(int j=1; j <= NO_STUDENT; j++)
   {
       cout << "Average mark for student " << j << " is " <<
student_average[j-1] << endl;
   }

   return 0;
}
```

Continued...

Continued...

(c) Write a class definition `TimeOfDay` that consists of the followings:

i. Two private members: `hour` and `minute`, both of type `int`.

ii. One public member function `void display()`. Write the definition to display the `hour` and `minute` in **hour:minute** format.

iii. One public member function `void set(int a, int b)`. Write the definition to set the `hour` and `minute`. Assume the first parameter is for `hour` and the second parameter is for `minute`.

[3 marks]

Continued...

Question 4

(a) What is the purpose of placing the "#include <fstream>" directive in your C++ program? [1 mark]

(b) Find the error (by stating the line number) from the program below. Then, fix the error. Hint that the line "`using namespace std;`" must not be used. [1 mark]

```
#include <iostream>                    // Line 1
int main()                             // Line 2
{                                      // Line 3
    cout << "no error" << "\n";        // Line 4
    return 0;                          // Line 5
}                                      // Line 6
```

(c) Given the program below. [3 mark]

```
#include <iostream>                    // Line 1
#include <vector>                      // Line 2
using namespace std;                   // Line 3
int main()                             // Line 4
{                                      // Line 5
    int array1[1000000];               // Line 6
    vector<int> array2(1000000);       // Line 7
    int* array3 = new int[1000000];    // Line 8
}                                      // Line 9
```

i. Which line number of the program causes error?
ii. Based on the given program, give two advantages of using vector over dynamic array.

i.

ii.

Continued...

(c) Complete the program below that consists of the followings: [5 marks]

- Asks the user to enter the row and column dimensions of the dynamic array
- Creates a two-dimensional dynamic array of type integer based on the entered row and column
- Asks the user to enter an integer for each element of the two-dimensional dynamic array
- Displays all the elements of the two-dimensional dynamic array
- Deletes the two-dimensional dynamic array

```
#include <iostream>
using namespace std;

int main()
{
    // Write your code from here
```

```
    // End your code here
    return 0;
}
```

End of Paper.